

## SCRIPT\_SQL.sql

```
1  /*Criação do banco de dados*/
2  CREATE DATABASE IF NOT EXISTS ACADEMIA;
3  USE ACADEMIA;
4
5  /*Tabela de benefícios*/
6  CREATE TABLE BENEFICIO (
7      BEN_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
8      BEN_STR_NOME VARCHAR(30) NOT NULL,
9      BEN_STR_DESC VARCHAR(500),
10     PRIMARY KEY (BEN_INT_ID)
11 );
12
13 /*Tabela de estados*/
14 CREATE TABLE ESTADO (
15     EST_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
16     EST_STR_DESC VARCHAR(50) NOT NULL,
17     EST_STR_SIGLA CHAR(2) NOT NULL UNIQUE,
18     PRIMARY KEY (EST_INT_ID)
19 );
20
21 /*Tabela de cidades*/
22 CREATE TABLE CIDADE (
23     CID_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
24     EST_INT_ID INT UNSIGNED NOT NULL,
25     CID_STR_DESCRICAO VARCHAR(50) NOT NULL,
26     PRIMARY KEY (CID_INT_ID),
27     FOREIGN KEY (EST_INT_ID)
28         REFERENCES ESTADO (EST_INT_ID)
29         ON DELETE CASCADE
30         ON UPDATE CASCADE
31 );
32
33 /*Tabela de endereços*/
34 CREATE TABLE ENDERECO (
35     END_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
36     CID_INT_ID INT UNSIGNED NOT NULL,
37     END_STR_LOGRADOURO VARCHAR(70) NOT NULL,
38     END_STR_NUMERO CHAR(5) NOT NULL,
39     END_STR_COMPLEMENTO VARCHAR(70),
40     END_STR_BAIRRO VARCHAR(50) NOT NULL,
41     END_STR_CEP CHAR(8) NOT NULL,
42     PRIMARY KEY (END_INT_ID),
43     FOREIGN KEY (CID_INT_ID)
44         REFERENCES CIDADE (CID_INT_ID)
45         ON DELETE CASCADE
46         ON UPDATE CASCADE
47 );
48
49 /*Tabela de clientes*/
50 CREATE TABLE CLIENTE (
51     CLI_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
```

```
52     END_INT_ID INT UNSIGNED,
53     CLI_STR_NOME VARCHAR(70) NOT NULL,
54     CLI_STR_SENHA VARCHAR(50) NOT NULL,
55     CLI_STR_CPF CHAR(11) NOT NULL UNIQUE,
56     CLI_STR_EMAIL VARCHAR(70) NOT NULL UNIQUE,
57     CLI_STR_TELEFONE CHAR(11) NOT NULL,
58     PRIMARY KEY (CLI_INT_ID),
59     FOREIGN KEY (END_INT_ID)
60         REFERENCES ENDERECO (END_INT_ID)
61         ON DELETE SET NULL
62         ON UPDATE CASCADE
63 );
64
65 /*Tabela de planos*/
66 CREATE TABLE PLANO (
67     PLA_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
68     PLA_STR_NOME VARCHAR(20) NOT NULL,
69     PLA_STR_PRECO DECIMAL(5, 2) NOT NULL,
70     PLA_STR_DESC VARCHAR(500),
71     PLA_STR_DURACAO VARCHAR(30) NOT NULL,
72     PRIMARY KEY (PLA_INT_ID)
73 );
74
75 /*Inserir planos iniciais*/
76 INSERT INTO PLANO (PLA_STR_NOME, PLA_STR_PRECO, PLA_STR_DESC, PLA_STR_DURACAO)
77 VALUES
78 ('Básico', 50.00, 'Treino inteligente e treinos em casa', 'Mensal'),
79 ('Plus', 80.00, 'PRO Gyms, treino inteligente e treinos em casa', 'Mensal'),
80 ('Black', 120.00, 'ELITE Gyms & Classes, PRO Gyms, treino inteligente e treinos em casa',
81     'Mensal');
82
83 /*Tabela de assinaturas */
84 CREATE TABLE ASSINATURA (
85     ASS_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
86     CLI_INT_ID INT UNSIGNED NOT NULL,
87     PLA_INT_ID INT UNSIGNED NOT NULL,
88     ASS_STR_DATA_INICIO DATE NOT NULL,
89     ASS_STR_DATA_FIM DATE,
90     ASS_STR_STATUS ENUM('Ativo', 'Cancelado', 'Pendente') NOT NULL DEFAULT 'Pendente',
91     PRIMARY KEY (ASS_INT_ID),
92     FOREIGN KEY (CLI_INT_ID)
93         REFERENCES CLIENTE (CLI_INT_ID)
94         ON DELETE CASCADE
95         ON UPDATE CASCADE,
96     FOREIGN KEY (PLA_INT_ID)
97         REFERENCES PLANO (PLA_INT_ID)
98         ON DELETE CASCADE
99         ON UPDATE CASCADE
100 );
101
102 /*Tabela de avaliações*/
103 CREATE TABLE AVALIACAO (
104     AVA_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
```

```
105     AVA_STR_ESTRELA CHAR(1) NOT NULL CHECK (AVA_STR_ESTRELA IN ('1', '2', '3', '4', '5')),
106     AVA_STR_DESC VARCHAR(500),
107     AVA_STR_DATA DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
108     PRIMARY KEY (AVA_INT_ID),
109     FOREIGN KEY (CLI_INT_ID)
110         REFERENCES CLIENTE (CLI_INT_ID)
111         ON DELETE CASCADE
112         ON UPDATE CASCADE
113 );
114
115
116 /*Tabela de benefícios vinculados aos planos*/
117 CREATE TABLE PLANO_BENEFICIO (
118     PB_INT_ID INT UNSIGNED NOT NULL AUTO_INCREMENT,
119     BEN_INT_ID INT UNSIGNED NOT NULL,
120     PLA_INT_ID INT UNSIGNED NOT NULL,
121     PRIMARY KEY (PB_INT_ID),
122     FOREIGN KEY (BEN_INT_ID)
123         REFERENCES BENEFICIO (BEN_INT_ID)
124         ON DELETE CASCADE
125         ON UPDATE CASCADE,
126     FOREIGN KEY (PLA_INT_ID)
127         REFERENCES PLANO (PLA_INT_ID)
128         ON DELETE CASCADE
129         ON UPDATE CASCADE,
130     UNIQUE KEY uk_plano_beneficio (BEN_INT_ID, PLA_INT_ID)
131 );
132
133 /* CRUD */
134 /* CLIENTE */
135 DELIMITER $$
136
137 CREATE PROCEDURE SP_CREATE_CLIENTE(
138     IN P_NOME VARCHAR(70),
139     IN P_EMAIL VARCHAR(70),
140     IN P_SENHA VARCHAR(50),
141     IN P_CPF CHAR(11),
142     IN P_TELEFONE CHAR(11),
143     IN P_ENDERECO_ID INT
144 )
145 BEGIN
146     DECLARE CLI_ID INT;
147
148     -- Inserir cliente
149     INSERT INTO CLIENTE (CLI_STR_NOME, CLI_STR_EMAIL, CLI_STR_SENHA, CLI_STR_CPF,
150     CLI_STR_TELEFONE, END_INT_ID)
151     VALUES (P_NOME, P_EMAIL, P_SENHA, P_CPF, P_TELEFONE, P_ENDERECO_ID);
152
153     -- Obter ID do cliente inserido
154     SET CLI_ID = LAST_INSERT_ID();
155
156     -- Retornar o ID do cliente
157     SELECT CLI_ID AS CLI_INT_ID;
158 END $$
```

```
158
159 DELIMITER ;
160
161 DELIMITER $$
162
163 CREATE PROCEDURE SP_DELETE_CLIENTE(
164     IN P_CLIENTE_ID INT
165 )
166 BEGIN
167     DELETE FROM CLIENTE WHERE CLI_INT_ID = P_CLIENTE_ID;
168 END $$
169
170 DELIMITER ;
171 DELIMITER $$
172
173 CREATE PROCEDURE SP_GET_ALL_CLIENTES()
174 BEGIN
175     SELECT * FROM CLIENTE;
176 END $$
177
178 DELIMITER ;
179
180 DELIMITER $$
181
182 CREATE PROCEDURE SP_GET_CLIENTE_BY_ID(
183     IN P_CLIENTE_ID INT
184 )
185 BEGIN
186     SELECT * FROM CLIENTE WHERE CLI_INT_ID = P_CLIENTE_ID;
187 END $$
188
189 DELIMITER ;
190
191 /* ENDEREÇO */
192
193 DELIMITER $$
194
195 CREATE PROCEDURE SP_CREATE_ENDERECO(
196     IN P_LOGRADOURO VARCHAR(70),
197     IN P_NUMERO CHAR(5),
198     IN P_COMPLEMENTO VARCHAR(70),
199     IN P_BAIRRO VARCHAR(50),
200     IN P_CEP CHAR(8),
201     IN P_CIDADE_ID INT
202 )
203 BEGIN
204     DECLARE END_ID INT;
205
206     INSERT INTO ENDERECO (END_STR_LOGRADOURO, END_STR_NUMERO, END_STR_COMPLEMENTO,
207     END_STR_BAIRRO, END_STR_CEP, CID_INT_ID)
208     VALUES (P_LOGRADOURO, P_NUMERO, P_COMPLEMENTO, P_BAIRRO, P_CEP, P_CIDADE_ID);
209
210     SET END_ID = LAST_INSERT_ID();
```

```
211     SELECT END_ID AS END_INT_ID;
212 END $$
213
214 DELIMITER ;
215 CALL SP_CREATE_ENDERECO('Rua Exemplo', '123', 'Apto 2', 'Centro', '12345678', 1);
216
217 DELIMITER $$
218
219 CREATE PROCEDURE SP_CREATE_ASSINATURA(
220     IN P_CLIENTE_ID INT,
221     IN P_PLANO_ID INT,
222     IN P_DATA_INICIO DATE,
223     IN P_STATUS ENUM('ATIVO', 'CANCELADO', 'PENDENTE')
224 )
225 BEGIN
226     INSERT INTO ASSINATURA (CLI_INT_ID, PLA_INT_ID, ASS_STR_DATA_INICIO, ASS_STR_STATUS)
227     VALUES (P_CLIENTE_ID, P_PLANO_ID, P_DATA_INICIO, P_STATUS);
228 END $$
229
230 DELIMITER ;
231
232 DELIMITER $$
233
234 CREATE PROCEDURE SP_UPDATE_ASSINATURA(
235     IN P_ASSINATURA_ID INT,
236     IN P_STATUS ENUM('ATIVO', 'CANCELADO', 'PENDENTE'),
237     IN P_DATA_FIM DATE
238 )
239 BEGIN
240     UPDATE ASSINATURA
241     SET ASS_STR_STATUS = P_STATUS, ASS_STR_DATA_FIM = P_DATA_FIM
242     WHERE ASS_INT_ID = P_ASSINATURA_ID;
243 END $$
244
245 DELIMITER ;
246
247 DELIMITER $$
248
249 CREATE PROCEDURE SP_GET_PLANOS()
250 BEGIN
251     SELECT * FROM PLANO;
252 END $$
253
254 DELIMITER ;
255
256 DELIMITER $$
257
258 CREATE PROCEDURE SP_GET_BENEFICIOS_BY_PLANO(
259     IN P_PLANO_ID INT
260 )
261 BEGIN
262     SELECT B.BEN_STR_NOME, B.BEN_STR_DESC
263     FROM PLANO_BENEFICIO PB
264     JOIN BENEFICIO B ON PB.BEN_INT_ID = B.BEN_INT_ID
```

```
265     WHERE PB.PLA_INT_ID = P_PLANO_ID;
266 END $$
267
268 DELIMITER ;
269
270 DELIMITER $$
271
272 CREATE PROCEDURE SP_CREATE_ESTADO (
273     IN P_EST_DESC VARCHAR(50),
274     IN P_EST_SIGLA CHAR(2)
275 )
276 BEGIN
277     DECLARE EST_ID INT;
278
279     SELECT EST_INT_ID INTO EST_ID
280     FROM ESTADO
281     WHERE EST_STR_SIGLA = P_EST_SIGLA;
282
283     IF EST_ID IS NULL THEN
284         INSERT INTO ESTADO (EST_STR_DESC, EST_STR_SIGLA)
285         VALUES (P_EST_DESC, P_EST_SIGLA);
286         SET EST_ID = LAST_INSERT_ID();
287     END IF;
288
289     SELECT EST_ID AS EST_INT_ID;
290 END $$
291
292 DELIMITER ;
293
294
295 DELIMITER $$
296
297 CREATE PROCEDURE SP_CREATE_CIDADE(
298     IN P_CIDADE_NOME VARCHAR(50),
299     IN P_ESTADO_ID INT
300 )
301 BEGIN
302     DECLARE CID_ID INT;
303
304     -- Verificar se a cidade já existe
305     SELECT CID_INT_ID INTO CID_ID
306     FROM CIDADE
307     WHERE CID_STR_DESCRICAO = P_CIDADE_NOME AND EST_INT_ID = P_ESTADO_ID
308     LIMIT 1;
309
310     -- Se não existir, insira a nova cidade
311     IF CID_ID IS NULL THEN
312         INSERT INTO CIDADE (CID_STR_DESCRICAO, EST_INT_ID)
313         VALUES (P_CIDADE_NOME, P_ESTADO_ID);
314         SET CID_ID = LAST_INSERT_ID();
315     END IF;
316
317     -- Retornar o ID da cidade
318     SELECT CID_ID AS CID_INT_ID;
```

```
319 | END $$  
320 |  
321 | DELIMITER ;  
322 |
```